МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Дніпровський національний університет залізничного транспорту імені академіка В. Лазаряна

Кафедра «Електронні обчислювальні машини»

«ДО ЗАХИСТУ»

Завідувач кафедри
_____ Жуковицький І. В.
(підпис) (ПІБ)
«____» _____ 20____р.

ДИПЛОМНА РОБОТА

на здобуття освітнього ступеня «магістр»

Галузь знань 12 Інформаційні технології

Спеціальність 125 Кібербезпека
(код) (повна назва)

Тема «Огляд та дослідження методів аутентифікації користувачів у веб-додатках»

Theme «Overview and research of user authentication methods in web applications»

Керівник дипломного проекту _____ _____ Заєць О.П.
(посада) (підпис) (ПІБ)

Консультант розділу з БЖД _____ _____ Музикін М. І.
(посада) (підпис) (ПІБ)

Нормоконтролер _____ _____ Шаповалов В. О
(посада) (підпис) (ПІБ)

Студент групи _____ _____ Мусієнко А.С.
(група) (підпис) (ПІБ)

Student Musienko Angelina
(family name)

Дніпро
2020

Дніпровський національний університет залізничного транспорту імені академіка В. Лазаряна

Факультет _____ кафедра _____
Спеціальність _____

«ЗАТВЕРДЖУЮ»
Завідувач кафедри

_____
(підпис)
«___»_____20_ р.

ЗАВДАННЯ

до дипломної роботи на здобуття освітнього ступеня _____
(освітнього ступеня)

студента групи _____ _____
(номер групи) (ПІБ)

1 Тема дипломної роботи _____
_____
_____
_____

затверджена наказом по університету від «___»_____20___ р. № _____.

2 Термін подання студентом закінченої роботи_____

3 Вихідні дані до дипломної роботи_____
_____
_____
_____

4 Зміст пояснювальної записки (перелік питань до розробки) _____
_____
_____
_____
_____

5 Перелік креслень (демонстраційного матеріалу) _____
_____
_____
_____
_____
_____
_____
_____
_____

6 Розділи та консультанти

Розділ	Консультант	Підпис, дата	
		завдання видав	завдання прийняв

КАЛЕНДАРНИЙ ПЛАН

Назва розділу	Термін виконання	Обсяг розділу, %

Дата видачі завдання: «___»_____ 20___ р.

Керівник дипломної роботи _____ _____
(підпис) (ПІБ)

Завдання прийняв до виконання _____ _____
(підпис) (ПІБ)

РЕФЕРАТ

Мусієнко А.С. Огляд та дослідження методів аутентифікації користувачів у веб-додатках. Дніпровський національний університет залізничного транспорту ім. акад. В. Лазаряна, кафедра електронних обчислювальних машин. Дипломна магістерська робота. 60 сторінок. 14 рисунків. 1 таблиць. 27 джерел.

У дипломній магістерській роботі виконано огляд та аналіз методів аутентифікації у веб-додатку. Брались до уваги методи аутентифікації за її класифікацією: аутентифікація за паролем, аутентифікація за сертифікатом, аутентифікація за одноразовим паролем, аутентифікація за ключами доступу, аутентифікація за токенами.

На початку магістерської дипломної роботи було вивчення методів аутентифікації окремо у великих кількостях інформації, що давав нам Інтернет, та книжки, які були опубліковані.

Було проведено аналіз методів аутентифікації за критеріями, такими як: затребуваності з боку користувача та розробника, складності користування цими методами аутентифікації, наявністю протоколу в методі, сфера використання методів аутентифікації, також були перераховані переваги та недоліки цих методів.

Останнім кроком, можна вважати, програмна реалізація методів аутентифікації за токенами. Цими методами стали метод аутентифікації OAuth 2.0 та WebAuthn. Після програмної реалізації цих методів було зроблено порівняння методів. Після отримання порівняння цих методів було зроблено висновок на основі отриманих результатів.

МЕТОД АУТЕНТИФІКАЦІЇ, OAUTH 2.0, WEBAUTHN, ТОКЕН, ПАРОЛЬ.

ABSTRACT

Musienko A.S. Review and research methods of user authentication in web applications. Dnipro National University of Railway Transport named after acad. V. Lazaryan, Department of Electronic Computers. Master's thesis. 60 pages. 14 drawings. 1 table. 27 sources.

In the master's thesis review and analysis of authentication methods in the web application. The methods of authentication according to its classification were taken into account: authentication by password, authentication by certificate, authentication by one-time password, authentication by access keys, authentication by tokens.

At the beginning of the master's thesis was the study of authentication methods separately in large amounts of information provided to us by the Internet, and books that have been published.

The analysis of authentication methods was performed according to criteria such as: demand from the user and developer, the complexity of using these authentication methods, the presence of a protocol in the method, the scope of authentication methods, and listed the advantages and disadvantages of these methods.

The last step, we can consider, is the software implementation of token authentication methods. These methods are the OAuth 2.0 and WebAuthn authentication methods. After software implementation of these methods, a comparison of methods was made. After comparing these methods, a conclusion was made based on the results obtained.

AUTHENTICATION METHOD, OAUTH 2.0, WEBAUTHN, TOKEN, PASSWORD.

ЗМІСТ

ВСТУП

Інтерне́т— всесвітня система сполучених комп'ютерних мереж, що базуються на комплекті Інтернет-протоколів. Інтернет також називають мережею мереж, бо він складається з мільйонів локальних і глобальних приватних, публічних, академічних, ділових і урядових мереж, пов'язаних між собою з використанням різноманітних дротових, оптичних і бездротових технологій.

Інтернет в Україні виник ще у 1990 році, коли в одному з київських офісів відбувся перше з’єднання з мережею Інтернету. Йшли роки, Інтернет, технології та суспільство змінювалось в найкращу сторону, тому вже на 1 січня 2020 року в Україні кількість користувачів вже перевищує 28 мільйонів 787 тисяч. З котрих фізичних осіб – 25 мільйонів 683 тисяч [1]. Саме з доступністю інтернету пов’язано плюсів та мінусів його використання. З позитивних рис можна виділити таке: корисність використання, не треба пам’ятати багато інформації в голові. З позитивних рис випливає негативне, а саме з’явились люди, яким потрібна ваша інформація.

На даний час кількість некваліфікованих веб-програмістів стрімко зростає. Разом з тим, високими темпами розвиваються загрози безпеки веб-сервісів.

Я вважаю, що саме з цих причин, захист інформації у web-додатках стає дуже важливою проблемою, для подальшого його вирішення.

Мета цієї дипломної роботи полягає у тому, щоб розглянути деякі методи аутентифікації у web-додатках, та проаналізувати який з цих методів більш ефективний.

1 ЗАГАЛЬНА ІНФОРМАЦІЯ

Процес реєстрації користувача в системі складається з трьох взаємопов'язаних, виконуваних послідовно процедур: ідентифікації, аутентифікації і авторизації.

Ідентифікація - це процедура розпізнавання суб'єкта за його ідентифікатором. У процесі реєстрації суб'єкт пред'являє системі свій ідентифікатор і вона перевіряє його наявність у своїй базі даних. Суб'єкти з відомими системі ідентифікаторами вважаються легальними (законними), інші суб'єкти відносяться до нелегальних.

Аутентифікація - процедура перевірки автентичності суб'єкта, що дозволяє достовірно переконатися в тому, що суб'єкт, який пред'явив свій ідентифікатор, насправді є саме тим суб'єктом, ідентифікатор якого він використовує. Для цього він повинен підтвердити факт володіння деякою інформацією, яка може бути доступна тільки йому одному (пароль, ключ і т.п.).

Авторизація - процедура надання суб'єкту певних прав доступу до ресурсів системи після проходження їм процедури аутентифікації. Для кожного суб'єкта в системі визначається набір прав, які він може використовувати при зверненні до її ресурсів [2].

Приведу приклад, якщо ви хочете зайти у закритий клуб читачів Т. Шевченка вас ідентифікують (а саме, спросять ваше ім’я та прізвище), аутентифікують (показати паспорт, або інший документ для звіру фотографії) і авторизують (перевірять, що ви знаходитесь у списку запрошених гостей). Зробимо аналогію з прикладом вище, та темою програмного забезпечення. Традиційно під ідентифікацією розуміють отримання вашого email та username; під аутентифікацією розуміють перевірку знання паролю, а під авторизацією – перевірку ролі в системі (рішення про надання доступу до запрошеного сайту).

Вже на даний час існують більш складні схеми аутентифікації та авторизації.

1.1 Аутентифікація за паролем

Цей метод грунтується на тому, що користувач повинен надати username і password для успішної ідентифікації і аутентифікації в системі. Пара username / password задається користувачем при його реєстрації в системі, при цьому в якості username може виступати адреса електронної пошти користувача.

Стосовно до веб-додатків, існує кілька стандартних протоколів для аутентифікації за паролем [3].

1.1.1 HTTP authentication

HTTP authentication - це протокол, описаний в стандартах HTTP 1.0 / 1.1. Працює наступним чином:

- При зверненні неавторизованого користувача до захищеного ресурсу сервер повертає «401 Unauthorized» і додає заголовок «WWW-Authenticate»
- Браузер при отриманні відповіді з заголовком «WWW-Authenticate» викидає форму для введення логіна і пароля. І в подальшому при зверненні до даного ресурсу передає заголовок «Authorization», де зберігаються дані користувача для аутентифікації [4].

Існують кілька схем http авторизації:

- Basic
- Digest
- NTLM
- Negotiate

1.1.1.1 Basic

Basic - найбільш проста схема, при якій username і password користувача передаються в заголовку Authorization в незашифрованому вигляді (base64-encoded). Однак при використанні HTTPS (HTTP over SSL) протоколу, є відносно безпечною [3].

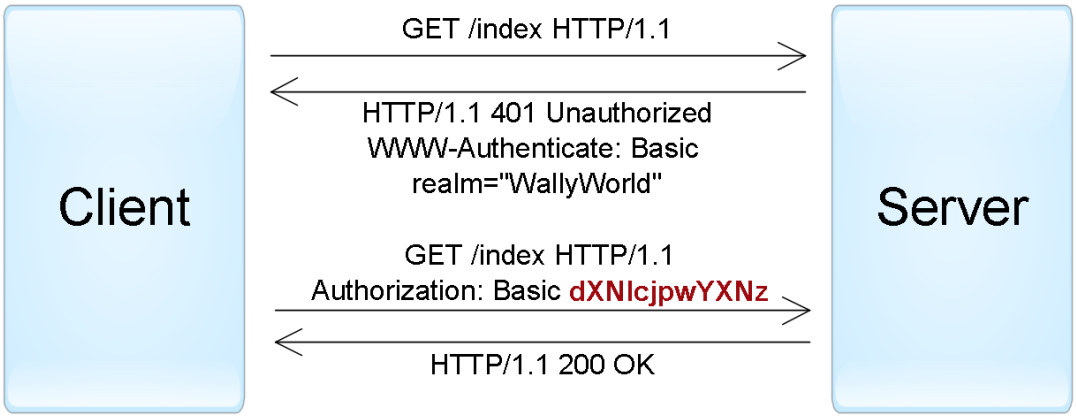


Рисунок 1 – Приклад HTTP аутентифікації з використанням Basic схеми.

1.1.1.2 Digest

Digest - challenge-response-схема, при якій сервер посилає унікальне значення nonce, а браузер передає MD5 хеш пароля користувача, обчислений з використанням зазначеного nonce. Більш безпечна альтернативв Basic схеми при незахищених з'єднаннях, але схильна до man-in-the-middle attacks (з заміною схеми на basic). Крім того, використання цієї схеми не дозволяє застосувати сучасні хеш-функції для зберігання паролів користувачів на сервері [3].

1.1.1.3 NTLM

NTLM (відома як Windows authentication) - також заснована на challenge-response підході, при якому пароль не передається в чистому вигляді. Ця схема не є стандартом HTTP, але підтримується більшістю браузерів і веб-серверів. Переважно використовується для аутентифікації користувачів Windows Active Directory в веб-додатках. Вразлива до pass-the-hash-атакам [3].

1.1.1.4 Negotiate

Negotiate - ще одна схема з сімейства Windows authentication, яка дозволяє клієнтові вибрати між NTLM і Kerberos аутентифікації. Kerberos - більш безпечний протокол, заснований на принципі Single Sign-On. Однак він може функціонувати, тільки якщо і клієнт, і сервер знаходяться в зоні intranet і є частиною домену Windows [3].

Важливим моментом, треба зазначити, що для користувачів HTTP - аутентифікації немає стандартної можливості вийти з додатку, для цього треба тільки закрити усі вікна у web-додатку.

1.1.2 Forms authentication

Аутентифікація за допомогою форм - це система аутентифікації загального призначення, заснована на двох концепціях. Перша з них - від мене вимагається залогуватись (login page), яка може засвідчити дійсність користувачів (зазвичай звіряючи комбінацію імені та пароля з базою даних або іншим сховищем даних). Друга - це механізм запобігання та відновлення контексту безпеки при кожному запиті (зазвичай із застосуванням cookie-набору). Таким чином, користувачеві знадобиться увійти тільки один раз.

Аутентифікація за допомогою форм заснована на квитках (також званих маркерами або їх ще називають session token). Це означає, що коли користувач реєструється, він отримує так званий квиток з базовою інформацією про себе. Інформація зберігається в зашифрованому cookie-наборі, який приєднується до відповіді, так що автоматично відправляється в кожному наступному запиті [5].

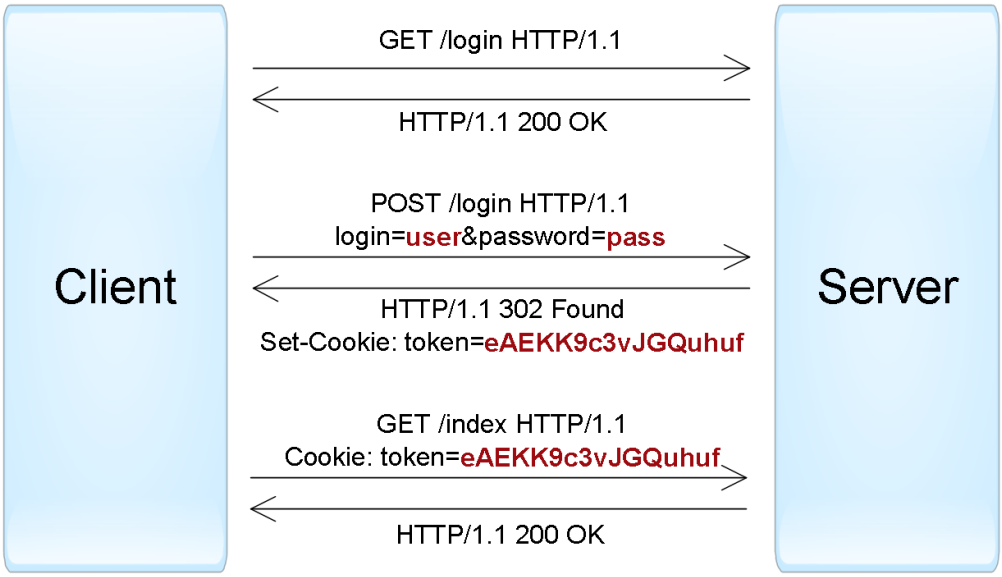


Рисунок 2 – Приклад роботи forms authentication

Необхідно розуміти, що перехоплення session token часто дає аналогічний рівень доступу, що і знання username / password. Тому всі комунікації між клієнтом і сервером у разі forms authentication повинні проводитися тільки по захищеному з'єднанню HTTPS [3].

1.1.3 Поширені уразливості і помилки реалізації

Аутентифікації по паролю вважається не дуже надійним способом, так як пароль часто можна підібрати, а користувачі схильні використовувати прості і однакові паролі в різних системах, або записувати їх на клаптиках паперу. Якщо зловмисник зміг з'ясувати пароль, то користувач часто про це не дізнається. Крім того, розробники додатків можуть допустити ряд концептуальних помилок, що спрощують злом облікових записів.

Нижче представлений список вразливостей, які найбільш часто зустрічаються в разі використання аутентифікації по паролю:

- Веб-додаток дозволяє користувачам створювати прості паролі.
- Веб-додаток не захищене від можливості перебору паролів (brute-force attacks).
- Веб-додаток саме генерує і поширює паролі користувачам, однак не вимагає зміни пароля після першого входу (тобто поточний пароль десь записаний), або взагалі не дає можливість змінити пароль, також не надає інформацію про зміну пароля.
- Веб-додаток допускає передачу паролів по незахищеному HTTP-з'єднання, або в рядку URL.
- Веб-додаток не використовує безпечні хеш-функції для зберігання паролів користувачів.
- Веб-додаток створює session tokens таким чином, що вони можуть бути підібрані або передбачені для інших користувачів.
- Веб-додаток допускає передачу session tokens по незахищеному HTTP-з'єднання, або в рядку URL.
- Веб-додаток не встановлює прапори HttpOnly і Secure для browser cookies, що містять session tokens.
- Веб-додаток не знищує сесії користувача після короткого періоду неактивності або не надає функцію виходу з аутентифицироваться сесії [3].

1.2 Аутентифікація за сертифікатом

Аутентифікація з застосуванням цифрових сертифікатів є альтернативою застосуванню паролів і видається природним рішенням в умовах, коли число користувачів мережі (нехай і потенційних) вимірюється мільйонами. В таких обставинах процедура попередньої реєстрації користувачів, пов'язана з призначенням і зберіганням їх паролів, стає вкрай обтяжливою, небезпечної, а іноді і просто нездійсненною. При наявності сертифікатів мережу, яка дає користувачеві доступ до своїх ресурсів, не зберігає ніякої інформації про своїх користувачів - вони її надають самі в своїх запитах у вигляді сертифікатів, що засвідчують особу користувачів. Сертифікати видаються спеціальними уповноваженими організаціями - центрами сертифікації (Certificate Authority, CA). Тому завдання зберігання секретної інформації (закритих ключів) покладається на самих користувачів, що робить це рішення набагато більш масштабується, ніж варіант з централізованою базою паролів [6].

На стороні клієнта сертифікат разом з закритим ключем можуть зберігатися в операційній системі, в браузері, в файлі, на окремому фізичному пристрої (smart card, USB token). Зазвичай закритий ключ додатково захищений паролем або PIN-кодом.

У веб-додатках традиційно використовують сертифікати стандарту X.509. Аутентифікація за допомогою X.509-сертифіката відбувається в момент з'єднання з сервером і є частиною протоколу SSL / TLS. Цей механізм також добре підтримується браузерами, які дозволяють користувачеві вибрати і застосувати сертифікат, якщо веб-сайт допускає такий спосіб аутентифікації [3].

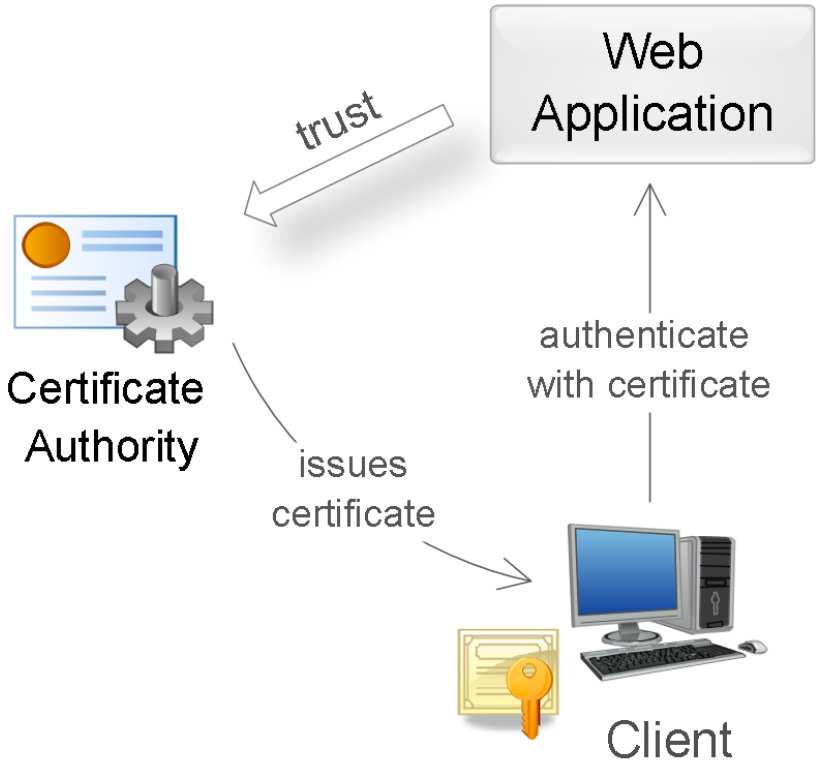


Рисунок 3 – Використання сертифіката для аутентифікації.

Важливо знати, що під час аутентифікації сервер проводить перевірку з наданим сертифікатом. Сервер використовує такі правила:

- Сертифікат повинен бути підписаним тільки довіреним certification authority;
- Перевірка часу дії сертифіката;
- Перевірка списків виключення, тобто сертифікат не повинен бути виключеним CA.

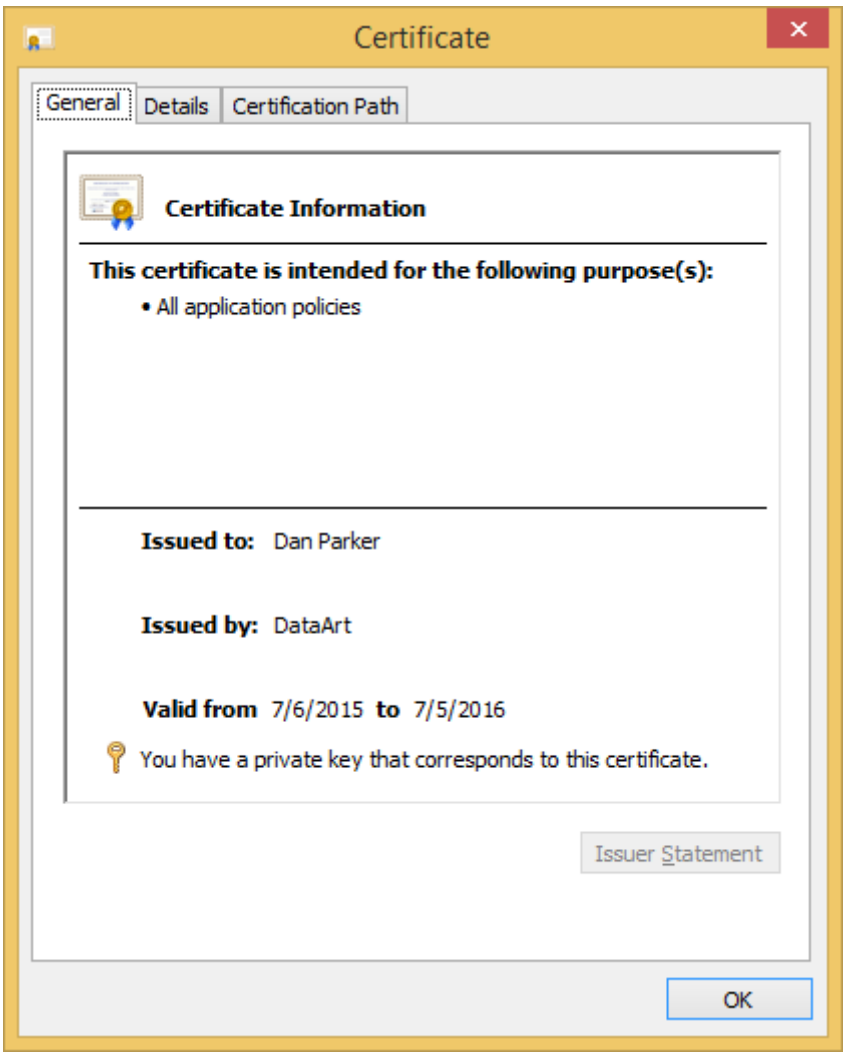


Рисунок 4 – Приклад X.509 сертифіката.

1.3 Аутентифікація за одноразовим паролем

One-Time-Password (OTP) – це ключове слово, дієве для одного процесу аутентифікації протягом обмеженого проміжку часу. Іншими словами, використання технології одноразових паролів (в теорії) дозволяє захистити якийсь «секрет» для повторного його використання при аутентифікації, крім того, цей «секрет» змінюється кожні 30-60 секунд [7].

Аутентифікація за одноразовими паролями зазвичай застосовується додатково до аутентифікації по паролів для реалізації two-factor authentication (2FA). У цій концепції користувачеві необхідно надати дані двох типів для входу в систему: щось, що він знає (наприклад, пароль), і щось, чим він володіє (наприклад, пристрій для генерації одноразових паролів). Наявність двох

факторів дозволяє в значній мірі збільшити рівень безпеки, що м. Б. затребуване для певних видів веб-додатків [3].

Існують багато видів джерел для створення одноразових паролів, роздивимось декілька варіантів:

- Апаратні або програмні токени, котрі можуть генерувати одноразові паролі на основі секретного ключа, які введені в апарат, та поточного часу. Секретний ключ знає користувач, також ключ зберігається на сервері, що дозволяє виконати перевірку паролів.

Приклади: апаратні - RSA SecurID, програмні – додаток Google Authenticator.

Рисунок 5 – апаратний токен RSA SecurID

- Випадково згенеровані коди, котрі передаються користувачеві через SMS або інший канал зв’язку.
- Роздруківка або scratch card зі списком раніше зроблений одноразових паролів.

1.4 Аутентифікація за ключами доступу

Цей спосіб найчастіше використовується для аутентифікації пристроїв, сервісів або інших додатків при зверненні до веб-сервісів. Тут в якості секрету застосовуються ключі доступу (access key, API key) - довгі унікальні рядки, що містять довільний набір символів, по суті замінюють собою комбінацію username / password.

Хороший приклад застосування аутентифікації по ключу - хмара Amazon Web Services. Припустимо, у користувача є веб-додаток, що дозволяє завантажувати і переглядати фотографії, і він хоче використовувати сервіс

Amazon S3 для зберігання файлів. В такому випадку, користувач через консоль AWS може створити ключ, що має обмежений доступ до хмари: тільки читання / запис його файлів в Amazon S3. Цей ключ в результаті можна застосувати для аутентифікації веб-додатки в хмарі AWS [3].

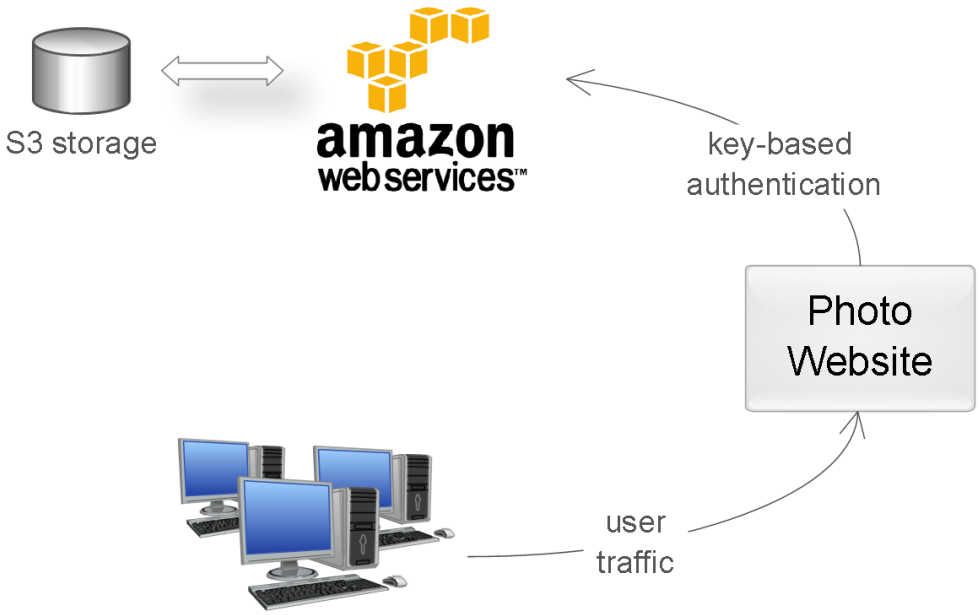


Рисунок 6 - Приклад застосування аутентифікації по ключу.

Використання ключів дозволяє уникнути передачі пароля користувача стороннім додаткам (в прикладі вище користувач зберіг в веб-додатку не свій пароль, а ключ доступу). Ключі мають значно більшу ентропією в порівнянні з паролями, тому їх практично неможливо підібрати. Крім того, якщо ключ був розкритий, це не призводить до компрометації основний облікового запису користувача - достатньо лише анулювати цей ключ і створити новий.

З технічної точки зору, тут не існує єдиного протоколу: ключі можуть передаватися в різних частинах HTTP-запиту: URL query, request body або HTTP header. Як і в випадку аутентифікації по паролю, найбільш оптимальний варіант - використання HTTP header. У деяких випадках використовують HTTP-схему Bearer для передачі токена в заголовку (Authorization: Bearer [token]). Щоб

уникнути перехоплення ключів, з'єднання з сервером має бути обов'язково захищене протоколом SSL / TLS [3].

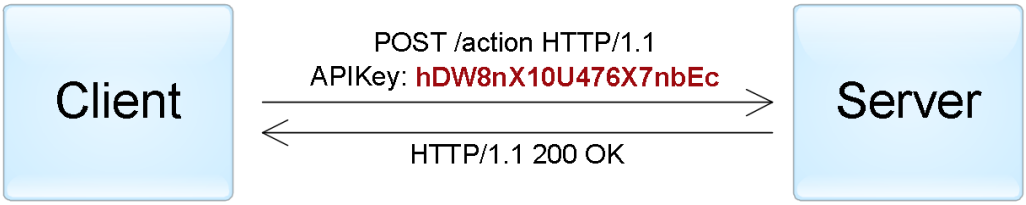


Рисунок 7 - Приклад аутентифікації по ключу доступу, переданого в HTTP заголовку.

Крім того, існують більш складні схеми аутентифікації по ключам для незахищених з'єднань. В цьому випадку, ключ зазвичай складається з двох частин: публічної і таємної. Публічна частина використовується для ідентифікації клієнта, а секретна частина дозволяє згенерувати підпис. Наприклад, за аналогією з digest authentication схемою, сервер може послати клієнту унікальне значення nonce або timestamp, а клієнт - повернути хеш або HMAC цього значення, обчислений з використанням секретної частини ключа. Це дозволяє уникнути передачі всього ключа в оригінальному вигляді і захищає від replay attacks [3].

1.5 Аутентифікація за токенами

Такий спосіб аутентифікації найчастіше застосовується при побудові розподілених систем Single Sign-On (SSO), де один додаток (service provider або relying party) делегує функцію аутентифікації користувачів іншому додатку (identity provider або authentication service). Типовий приклад цього способу - вхід в додаток через обліковий запис в соціальних мережах. Тут соціальні мережі є сервісами аутентифікації, а додаток довіряє функцію аутентифікації користувачів соціальних мереж [3].

Весь процес виглядає наступним чином:

- Клієнт аутентифікується в identity provider одним із способів, специфічним для нього (пароль, ключ доступу, сертифікат, Kerberos, і т.д.).
- Клієнт просить identity provider надати йому токен для конкретного SP-додатки. Identity provider генерує токен і відправляє його клієнту.
- Клієнт аутентифікується в SP-додатку за допомогою цього токена.

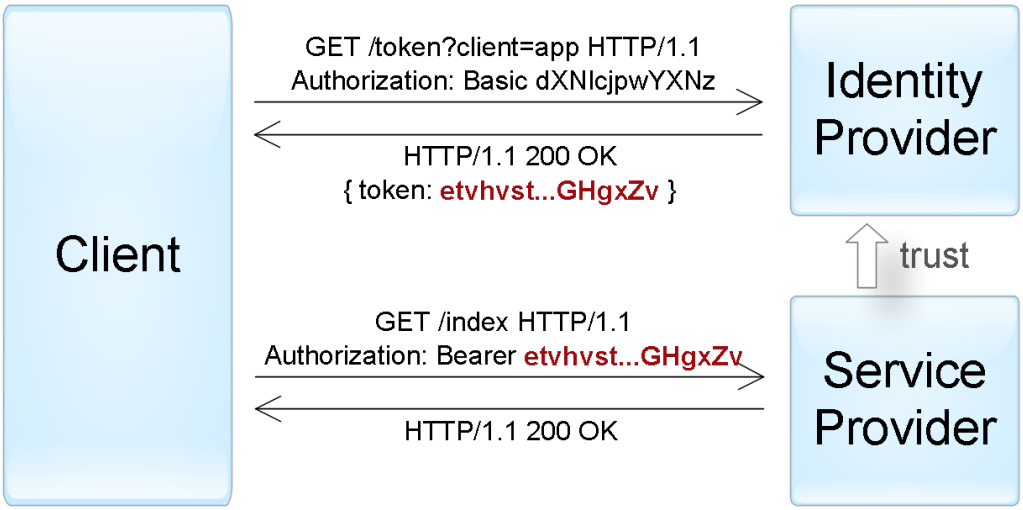


Рисунок 8 - Приклад аутентифікації «активного» клієнта за допомогою токена, переданого за допомогою Bearer схеми.

Процес, описаний вище, відображає механізм аутентифікації активного клієнта, тобто такого, який може виконувати запрограмовану послідовність дій (наприклад, iOS / Android програми). Браузер же - пасивний клієнт в тому сенсі, що він тільки може відображати сторінки, запитані користувачем. В цьому випадку аутентифікація досягається за допомогою автоматичного перенаправлення браузера між веб-додатками identity provider і service provider [3].

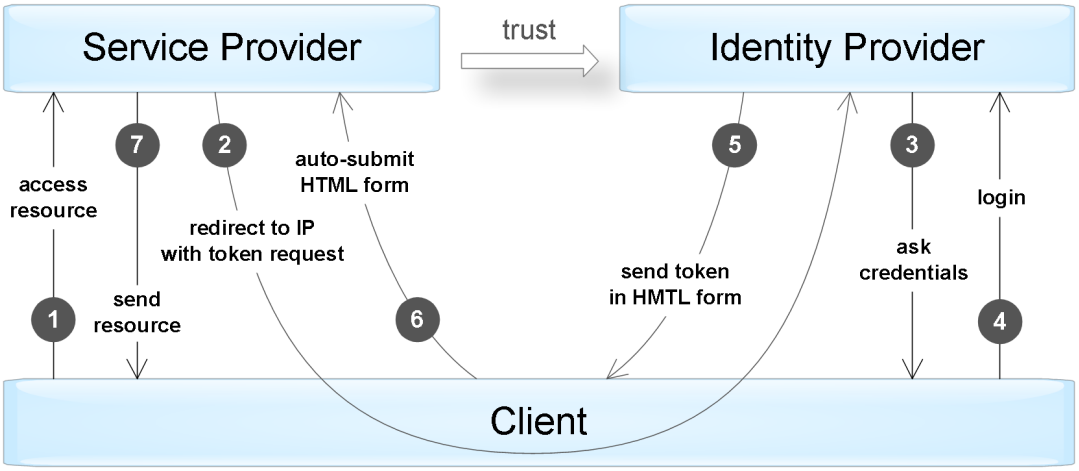


Рисунок 9 - Приклад аутентифікації «пасивного» клієнта за допомогою перенаправлення запитів.

Існує кілька стандартів, в точності що визначають протокол взаємодії між клієнтами (активними і пасивними) і IP / SP-додатками і формат підтримуваних токенів. Серед найбільш популярних стандартів - OAuth, OpenID Connect, SAML, і WS-Federation. Роздивимось поближче з цими стандартами.

1.5.1 OAuth

Це різновид єдиної точки входу зі спрощенням процесу реєстрації / входу користувача в вашу програму. Використовується при реєстрації / вході в додаток через соціальні мережі [8].

OAuth 1.0 (2010) дозволяє користувачеві вирішувати з додатком отримувати обмежений доступ на третьесторонніх серверах (third-party server), які довіряють засвідчувальному центру. OAuth 2.0 (2012) робить те ж саме, що і OAuth 1.0, але тільки протокол істотно змінився і став простіше [9].

OAuth 2.0 - протокол авторизації, що дозволяє видати одному сервісу (з додатком) права на доступ до ресурсів користувача на іншому сервісі. Протокол позбавляє від необхідності довіряти додатком логін і пароль, а також дозволяє видавати обмежений набір прав, а не все відразу [10].

Загальна схема роботи програми, що використовує OAuth, така:

- отримання авторизації
- звернення до захищених ресурсів

Результатом авторизації є access token - якийсь ключ (зазвичай просто набір символів), пред'явлення якого є пропуском до захищених ресурсів. Звернення до них в найпростішому випадку відбувається по HTTPS із зазначенням в заголовках або в якості одного з параметрів отриманого access token'a.

У протоколі описано кілька варіантів авторизації, що підходять для різних ситуацій:

- авторизація для додатків, що мають серверну частину (найчастіше, це сайти і веб-додатки)
- авторизація для повністю клієнтських додатків (мобільні і desktop-додатки)
- авторизація за логіном і паролем
- відновлення попередньої авторизації

Перевага: користувачі можуть увійти в ваше додаток одним кліком, якщо у них є аккаунт в одній з соцмереж. Їм не потрібно пам'ятати логіни і паролі. Це сильно покращує досвід використання вашого застосування. Вам як розробнику не потрібно хвилюватися про безпеку призначених для користувача даних і думати про перевірку адрес пошти - вони вже перевірені соцмережами. Крім того, в соцмережах вже є механізми відновлення пароля [8].

Мінуси OAuth 2.0

OAuth 2.0 - розвивається стандарт. Це означає, що специфікація ще не усталилася і постійно змінюється, іноді досить помітно. Так, що якщо ви вирішили підтримати стандарт прямо зараз, приготуйтеся до того, що його підтримку доведеться підпилювати в міру зміни специфікації. З іншого боку, це також означає, що ви можете взяти участь в процесі написання стандарту і внести в нього свої ідеї.

Безпека OAuth 2.0 багато в чому заснована на SSL. Це сильно спрощує життя розробникам, але вимагає додаткових обчислювальних ресурсів і адміністрування. Це може бути суттєвим питанням в високо навантажених проектах [10].

1.5.2 OpenID Connect

OpenID Connect (OIDC) - це тонкий шар поверх OAuth 2.0, який додає відомості про логін і профілі користувача, який увійшов в обліковий запис. Організацію логін-сесії часто називають аутентификацией [authentication], а інформацію про користувача, який увійшов в систему (тобто про Resource Owner'e), - особистими даними [identity]. Якщо Authorization Server підтримує OIDC, його іноді називають постачальником особистих даних [identity provider], оскільки він надає Client'у інформацію про Resource Owner'e.

OpenID Connect дозволяє реалізовувати сценарії, коли єдиний логін можна використовувати в безлічі додатків, - цей підхід також відомий як single sign-on (SSO). Наприклад, додаток може підтримувати SSO-інтеграцію з соціальними мережами, такими як Facebook або Twitter, дозволяючи користувачам використовувати обліковий запис, який у них вже є і яку вони вважають за краще використовувати [11].

1.5.3 SAML

Мова розмітки SAML (Security Assertion Markup Language) являє собою відкритий стандарт на основі XML, який призначений для обміну даними аутентифікації і авторизації між сторонами процесу. Що став стандартом з 2002 року, SAML є розробкою Технічного комітету з сервісами безпеки (Security Services Technical Committee), який працює при організації OASIS, що займається просуванням стандартів для роботи зі структурованою інформацією. За допомогою протоколу SAML користувачі можуть отримувати доступ до безлічі своїх хмарних додатків, вказуючи всього один логін і пароль. Такий підхід отримав назву «федерації посвідчень», оскільки замість запам'ятовування чималої кількості логінів і паролів до кожного з додатком, користувачеві необхідно пам'ятати лише одну таку пару. При федерації посвідчень єдина система, що підтримує протокол SAML і отримала назву довіреної постачальника посвідчень (Identity Provider, IdP), проводить аутентифікацію користувачів, при цьому хмарні додатки «перекидають» процес аутентифікації на цю IdP систему щоразу при спробі користувача отримати до них доступ.

Сервіс аутентифікації, який виступає в якості постачальника посвідчень, отримує призначені для користувача облікові дані і повертає відповідь того хмарного додатком, до якого здійснюється доступ. Ця відповідь отримав назву SAML підтвердження. Залежно від вмісту SAML підтвердження хмарне додаток або приймає, або відмовляє користувачеві в доступі. Якщо SAML підтвердження містить позитивну відповідь, то користувач входить в систему [12].

1.5.4 WebAuthn

Такі компанії як Microsoft, Yahoo, Amazon замислюються про використання безпарольному методів аутентифікації і про повну відмову від використання паролів в своїх сервісах.

У березні 2019 року W3C випустив першу версію стандарту, який описує браузерні JS API, що дозволяє взаємодіяти з електронними ключами. Стандарт отримав статус рекомендації і назва Web Authentication: API для доступу до облікових даних з використанням відкритого ключа: рівень перший (Web Authentication: An API for accessing Public Key Credentials Level 1) - скорочено WebAuthn.

Виділяються наступні основні учасники процесу аутентифікації з використанням WebAuthn:

- клієнт (WebAuthn Client) - браузер, що має підтримку WebAuthn API;
- Web-додаток - додаток, запущене на клієнті, що використовує WebAuthn API для взаємодії з обліковими даними;
- облікові дані (Public Key Credential) - пара з відкритого і закритого криптографічних ключів, які зв'язуються з профілем користувача;
- аутентифікатор (Authenticator) - пристрій або програма - створює облікові дані користувача і підписує цими обліковими даними запити від перевіряє боку (інша назва - електронний ключ);
- перевіряє сторона (WebAuthn Relying Party) - web-сервер - зберігає відкритий ключ, пов'язаний з профілем користувача, перевіряє коректність підпису своїх запитів закритим ключем, що зберігається в аутентифікаторі.

API WebAuthn дозволяє виробляти всього лише дві операції. Воно дозволяє створювати нові облікові дані і підписувати запити від сервера вже створеними обліковими даними [8].

2 АНАЛІЗ МЕТОДІВ АУТЕНТИФІКАЦІЇ ЗА ДЕЯКИМИ КРИТЕРІЯМИ

З приходом комп'ютерних технологій в повсякденне життя захист даних, переданих і збережених в мережі, стала необхідністю. Поряд з апаратними та програмними складовими, система захисту даних обов'язково повинна включати і кошти аутентифікації користувачів, які зможуть запобігти несанкціонованому доступу до акаунтів і облікових записів.

На сам перед кожен розробник або користувач вибирає метод аутентифікації за своїми критеріями, але на сам перед вибирають прості та швидкі методи аутентифікації. Але не можна казати, що вони більш незахищені, нажаль. Тому і мета цього дипломного проекту це визначення який ж з методів аутентифікації більш надійний для використання. Нажаль, повної відповіді немає, через те що методи аутентифікації не можуть збігатись. Деякі методи більш захищені, але вони програють своєю складністю користування, а деякі навпаки, більш прості та швидкі, але програють своєю незахищеністю.

2.1 Затребуваність з боку користувача

Під затребуваністю користувача необхідно розуміти, затребуваність для користування з боку користувача, для мене як не тільки розробника, але також у реальному житті – користувача, я розподілила на такі категорії:

- Висока – найвисокий рівень користуванням користувачів (дивись таблиця 1).
- Середня – середній рівень користування (дивись таблиця 1).
- Низька – низький рівень користуванням (дивись таблиця 1).

Але чому ж я так розподілила? Я враховувала кількість інформації по тому, або іншому методу авторизації у веб – додатках, а також врахувала складність користування, саме з боку користувача, а не розробника.

Аутентифікація за допомогою паролю – затребуваність висока. З боку користувача, цей варіант найлегший, тому що він більш доступний. Користувачі вже звикли до того, що на будь якому сайті можна просто ввести логін та пароль

і все, ви вже увійшли. Немає нічого складного, як для школяра, так і для пенсіонера. Враховуючи, що майже кожен користувач не буде використовувати дуже складні паролі, які будуть складатися з цифр, великих букв та малих букв, ні, для звичайного користувача це надзвичайно складно. Звичайний користувач буде використовувати все щоб зробити пароль для легкого запам’ятовування або ж вони просто використають папірець для запису. Але про це ми поговоримо трохи пізніше (дивись пункт 2.3)

Аутентифікація за допомогою сертифіката/ключами доступу – затребуваність низька, через те, що не для кожного користувача буде просто скористатися сертифікатом/ключем, це не брати до уваги отримання сертифікату/ключа.

Аутентифікація за одноразовим паролем – затребуваність середня, взявши до уваги те, що для отримання одноразового паролю частіше за все потрібен пристрій, який буде генерувати одноразовий пароль. По Україні, та інших країнах, ця річ досить дорога. Тобто, для простого користувача інтернетом, для хатніх справ, таких як, подивитись серіали на якомусь сайті – це непотрібно. Але, якщо дивитись з боку підприємця, котрий зберігаю свою інформацію та інформацію його підприємства – ця річ дуже знадобиться. Тому і мій вибір впав на затребуваність середню.

Аутентифікація за токеном – затребуваність висока. Під цим я розумію використання стандартів OAuth або OpenID Connect. Я вважаю, що для користувача це надзвичайно легко. Запам’ятати лише логін та пароль від одного аккаунту, та робити вхід використати ці данні, не переходячи на інші вікна. Програма все зробить за вас.

2.2 Затребуваність з боку розробника

Під затребуваності з боку розробника, необхідно розуміти затребуваність розробника для створення або використання того або іншого методу у своєму

веб-додатку в якості забезпечення зберігання інформації користувача, розподіливши на такі категорії:

- Низька – під низькою затребуваність треба розуміти, що для розробника не варто використовувати та марнувати час на цю аутентифікацію.
- Середня – під середньою затребуваністю треба розуміти, що розробник витратить багато часу, але цей час буде не в пусту, через те що цей метод досить захищений від хакерських атак.
- Висока - під високою затребуваністю треба розуміти, що для розробника це гарний вибір використати у своєму веб – додатку цей метод.

Отже, чому я так розподілила указано нижче.

Аутентифікація за паролем – низька, через те, що, по-перше, так, для розробника не буде важно написати код для пропуску лише з використанням логіну та паролю, але як я вже казала (дивись пункт 2.1), що майже кожен простий користувач буде використовувати легко доступні паролі або використовувати папірець. Великий недолік цьому це хакери, які зможуть легко скористатися нагодою та захватити інформацію користувача. Задача ж розробника це максимально захистити користувачів від хакерів. Тому на мою думку, аутентифікація за паролем не важить витраченого часу написання коду.

Аутентифікація за допомогою сертифіката/колюча доступу – середня, на мою думку, для розробника багато роботи щоб реалізувати на своєму сайті (якщо під авторизацією за допомогою сертифіката, розуміти SSL – сертифікат).

Щоб реалізувати процес авторизації за клієнтськими SSL сертифікатами, необхідно виконати наступне:

1. Отримати свій власний довірений сертифікат Certificate Authority, щоб потім з його допомогою підписувати і верифікувати клієнтські сертифікати.
2. Створити клієнтські сертифікати, які потім будуть передаватися клієнтам. Такі сертифікати повинні бути підписані довіреною сертифікатом.

3. Налаштувати сервер, щоб він належним чином запитував і верифікувати клієнтські SSL-сертифікати [18].

Але використовуючи авторизацію за допомогою сертифіката/ключа доступу є й свої переваги. Перевага в тому що цей метод досить захищений від хакерів.

Аутентифікація за одноразовим паролем – висока. З точки зору розробника, цей метод досить захищений та простий. Якщо під одноразовим паролем будемо розуміти створену розробником базу з одноразовими паролями, які будуть діяти певний період часу, наприклад 2 хвилини. Користувач за цей час повинен або увійти до сайту, або змінити на свій пароль. Це вже вибирає сам розробник.

Аутентифікація за токеном – висока. На мою думку, це найефективніший метод не тільки для користувача, а також і для розробника (під методом беру до уваги стандарт OAuth 2.0).

Перед тим, як почати використовувати OAuth в вашому додатку, вам необхідно зареєструвати своє додатки в сервісі. Це робиться шляхом реєстрації в розділі "developer" або "API" сайту сервісу, де вам необхідно надати наступну інформацію (можливо, включаючи деякі деталі про вашому додатку):

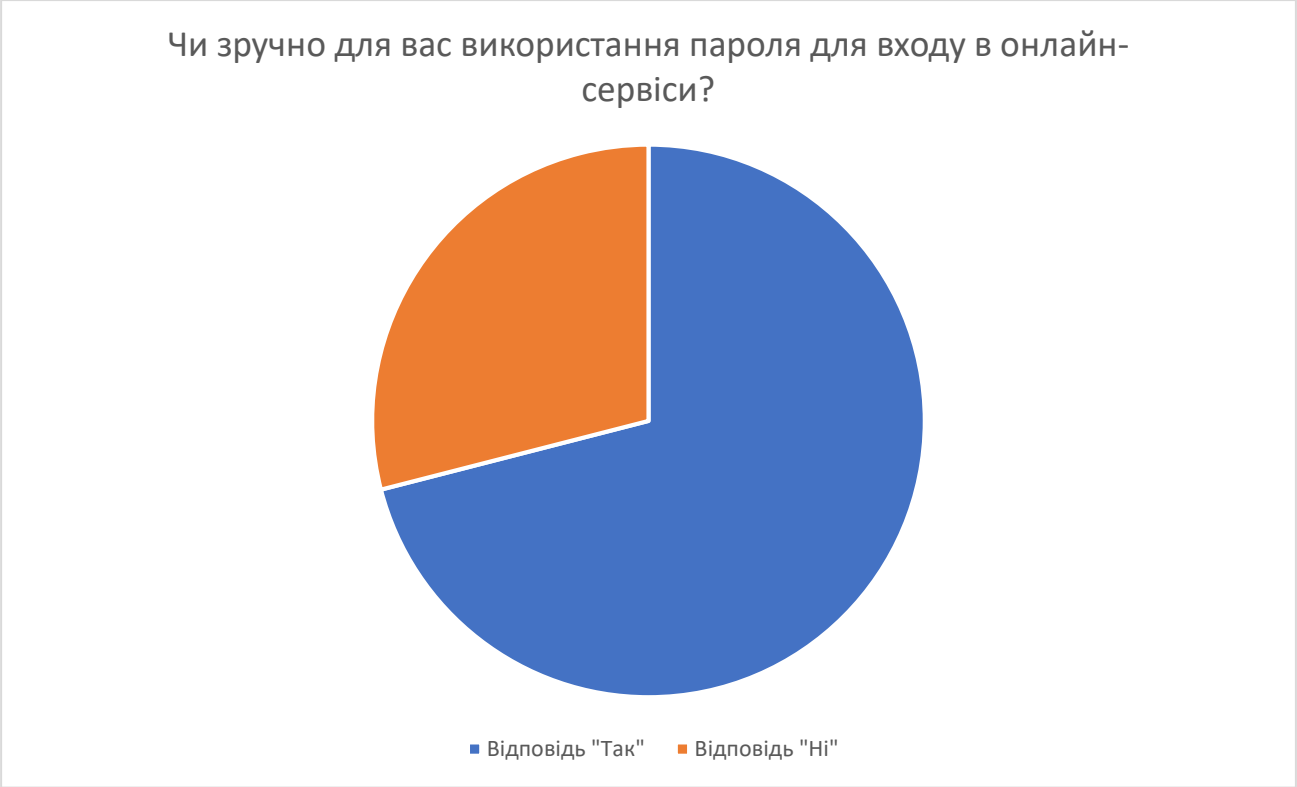
- Назва додатка
- Сайт додатки
- Redirect URL або callback URL [19].

2.3 Складність користування

Аутентифікація за паролем – на мою думку, для користувача немає ніякої складності, якщо не вважати той факт, як я вже казала вище (дивись пункт 2.1), що користувач не вибере доволі легкий пароль.

Згідно з опитуванням, яке провели серед студентів СПБГТИ (ТУ) (Санкт-Петербурзький державний технологічний інститут (технічний університет)) у студентів було питання «Чи зручно для вас використання пароля для входу в

онлайн-сервіси?». За результатами опитування 71% студентів вважають пароль зручним методом аутентифікації, коли 29% вважати незручним [20].



Діаграма 1 – Результати опитування серед студентів СПБГТИ (ТУ)

Також згідно з аналізом бази даних компанії з управління паролями NordPass, користувачі як і раніше використовують найпростіші паролі, які легко зламати.

Перевіривши майже 275,7 мільйона паролів, NordPass опублікував список найбільш часто використовуваних паролів для онлайн-акаунтів в 2020 році. Слід подивитись, які ж паролі використовують користувачі [21].

Більш короткий пароль «12345» посідав перше місце в минулому році. Але більш 188000 користувачів вибрали його в цьому році, що дозволило зайняти йому восьме місце. Обидва пароля можна зламати менш ніж за секунду.

NordPass заявив, що менше половини паролів в списку 2020 року є новими. Дослідження показують, що через зручності люди використовують прості і легко запам'ятовуються паролі, а також нецензурні слова, числа, імена та їжу.

В першу десятку (з 17 на 6 місце) піднявся пароль «111111», а пароль «123123» піднявся з 18 на 7 сходинку. Пароль «picture1» став новим в цьому році і відразу зайняв третє місце. Четверте місце зайняв «password». Десяте місце дісталося новинці «senha». На португальською мовою це слово означає «пароль».

Популярна комбінація «qwerty» займає 12 місце. Джерело додало, що деякі паролі з першої десятки часто зламуються за секунди, а на «picture1» йде близько 3 годин.

Position	Password	Number of users	Time to crack it	Times exposed
1. ↑ (2)	123456	2,543,285	Less than a second	23,597,311
2. ↑ (3)	123456789	961,435	Less than a second	7,870,694
3. (new)	picture1	371,612	3 Hours	11,190
4. ↑ (5)	password	360,467	Less than a second	3,759,315
5. ↑ (6)	12345678	322,187	Less than a second	2,944,615
6. ↑ (17)	111111	230,507	Less than a second	3,124,368
7. ↑ (18)	123123	189,327	Less than a second	2,238,694
8. ↓ (1)	12345	188,268	Less than a second	2,389,787
9. ↑ (11)	1234567890	171,724	Less than a second	2,264,884
10. (new)	senha	167,728	10 Seconds	8,213
11. ↑ (12)	1234567	165,909	Less than a second	2,516,606
12. ↓ (10)	qwerty	156,765	Less than a second	3,946,737
13. ↑ (16)	abc123	151,804	Less than a second	2,877,689
14. (new)	Million2	143,664	3 Hours	162,609
15. ↑ (28)	000000	122,982	Less than a second	1,959,780
16. ↓ (15)	1234	112,297	Less than a second	1,296,186
17. ↓ (14)	iloveyou	106,327	Less than a second	1,645,337
18. (new)	aaron431	90,256	3 Hours	30,576
19. ↑ (29)	password1	87,556	Less than a second	2,418,984
20. (new)	qqww1122	85,476	52 Minutes	122,481

Рисунок 10 – Двадцятка більш поширених паролей серед користувачів у 2020 році.

Аутентифікація за допомогою сертифіката/ключа доступу – складність буде тільки в тому, що сертифікат/ключ доступу не так легко отримати. Ще складність буде у тому, що у сертифікат/ключ доступу є термін придатності. Буде ще гірше, коли користувач втратить свій сертифікат/ключ доступу.

Аутентифікація за одноразовим паролем - складність для користування пов'язана з поступовим неузгодженням внутрішніх годинників та апаратного ключа. Тому теоретично можна припустити, що за час розсинхронизації, коли одноразовий пароль може стати многоразовим, хакер може скористатися нагодою.

Генерація одноразових паролів може виконуватися або програмно, або апаратно. Апаратні реалізацій систем доступу на основі одноразових паролів називають апаратними ключами. Вони являють собою мініатюрні пристрої з вбудованим мікропроцесором, схожі або на звичайні пластикові картки, використовувані для доступу до банкоматів, або на кишенькові калькулятори, які мають клавіатуру і маленьке дисплейне вікно [16].

Розглянемо схему використання апаратних ключів, в основі якої лежить синхронізація за часом. Цей популярний алгоритм аутентифікації був розроблений компанією Security Dynamics.

Ідея методу полягає в тому, що апаратний ключ і аутентифицирующей сервер обчислюють деяке значення пр-одному і тому ж алгоритму. Алгоритм має два параметри:

- розділяється секретний ключ, який представляє собою 64-розрядне число, унікально призначається кожному користувачеві і яке зберігається як в апаратній ключі, так і в базі даних сервера аутентифікації;
- значення поточного часу.

Потенційною проблемою цієї схеми є тимчасова синхронізація сервера і апаратного ключа (ясно, що питання узгодження часових поясів вирішується просто). Набагато складніше йде справа з поступовим неузгодженістю внутрішнього годинника сервера і апаратного ключа, тим більше що потенційно

апаратний ключ може працювати кілька років. Компанія Security Dynamics вирішує цю проблему двома способами. По-перше, при виробництві апаратного ключа вимірюється відхилення частоти його таймера від номіналу. Далі ця величина враховується в вигляді параметра алгоритму сервера. По-друге, сервер відстежує коди, що генеруються конкретним апаратним ключем, і якщо таймер даного ключа постійно поспішає або відстає, то сервер динамічно підлаштовується під нього [16].

Існує ще одна проблема, пов'язана зі схемою часової синхронізації. Одноразовий пароль, що генерується апаратним ключем, дійсний протягом деякого інтервалу часу (від декількох десятків секунд до декількох десятків хвилин), тобто протягом цього часу одноразовий пароль, по суті, є багаторазовим. Тому теоретично можливо, що дуже моторний хакер зможе перехопити PIN-код і одноразовий пароль з тим, щоб також отримати доступ в мережу протягом цього інтервалу [16].

Аутентифікація за токеном – на мою думку, це найлегший спосіб для користувача, тому що дозволяє користувачеві перемикатись між різними додатками без повторної аутентифікації.

2.4 Наявність протоколу

Аутентифікація за паролем – використовує HTTP, Forms.

Аутентифікація за допомогою сертифіката/ключа доступу – використовує SSL/TLS.

Аутентифікація за одноразовим паролем – використовує Forms.

Аутентифікація за токеном - використовує SAML, WS-Federation, OAuth, OpenID Connect.

2.5 Сфера використання методу

Аутентифікація за паролем – давайте будемо чесними, та признаємо, що хоч парольна аутентифікація не надійна, її використовують усюди. Від WEB-сайтів, мережеві служби до авторизації в смартфонах/планшетах.

Аутентифікація за допомогою сертифіката/ключа доступу – більш за все використовуються у електронній комерції або в будь-який інший сфері, де потрібна машинна аутентифікація або необхідні безпечні з'єднання [17].

Аутентифікація за одноразовим паролем – частіше використовують у мобільних банках, Web-сайтах, OTP-токени [13].

Аутентифікація за токеном - вхід в додаток через обліковий запис в соціальних мережах, систем накопичувальних знижок до кредитних і дебетових карт, студентських квитків, телефонів стандарту GSM (знайома всім SIM-карта, по суті та ж смарт-карта, тільки без зайвого пластика і зі спеціальним ПО), проїзних квитків [14].

2.6 Переваги та недоліки методів аутентифікації

2.6.1 Переваги

Аутентифікація за паролем:

Головна перевага парольної ідентифікації я вже казала (дивись пункт 2.1) - це простота реалізації й використання. Крім того, введення парольної ідентифікації не вимагає зовсім ніяких витрат: даний процес реалізований у всіх програмних продуктах, що є в продажу. Таким чином, система захисту інформації виявляється гранично простою і доступною.

Аутентифікація за допомогою сертифіката/ключа доступу:

Головна перевага цього методу це надійність. Також те, що сертифікат/ключ доступу є одним: користувачеві для повноцінного функціонування в інтернет-просторі досить одного сертифіката, що радикально спрощує доступ до облікових записів, і позбавляє від необхідності пам'ятати десятки паролів.

Аутентифікація за одноразовим паролем:

Перевага є в тому, що одноразовий дуже складно перехватити хакером, тому можна зробити висновок, що цей метод є надійнішим.

Аутентифікація за токеном:

Головною перевагою є простота, кожна людина дуже легко и швидко зрозуміє як користуватись. Також, перевагою є те, що користувачеві не потрібно перемикатись між різними додатками без повторної аутентифікації, треба запам’ятати лише один пароль и все.

2.6.2 Недоліки

Аутентифікація за паролем:

Найголовнішим недоліком слід вважати людський фактор. Користувачі не зі всією відповідальністю відносяться до створення паролю надійного (дивись пункт 2.3), нажаль на цьому витоки інформації дуже великі.

Аутентифікація за допомогою сертифіката/ключа доступу:

Одним з недоліків слід вважати те, що користувачеві складно отримати сертифікат/ключ доступу, через те, що зараз дуже багато сертифікованих центрів, але це не означає, що вони усі дійсні. Можна легко натрапити на мошенників. Ще одним з недоліків можна вважати, що сертифікат/ключ доступу дається не на все життя, а на деякий час.

Аутентифікація за одноразовим паролем:

Головний недолік це доволі висока ціна додаткового пристрою, не кожен користувач зможе використати. Ще один недолік є те, що бувають розсинхранизації часу з сервером, що є для хакеру нагодою скористатись моментом.

Аутентифікація за токеном:

Нажаль, аутентифікація за токеном має свої недоліки, це те що, можливий сценарій «людини по середині». Метод компрометації каналу зв'язку, при якому зламник, під'єднавшись до каналу між контрагентами, здійснює активне

втручання в протокол передачі, видаляючи, спотворюючи інформацію або нав'язуючи хибну.

Таблиця 1 – Порівняльний аналіз методів аутентифікації за критеріями

Метод аут. / Критерія	Аутентифікація за паролем	Аутентифікація за сертифікатом/ключем	Аутентифікація за одноразовим паролем	Аутентифікація за токенами
Затребуваність (користувач)	Висока	Низька	Середня	Висока
Затребуваність (розробник)	Низька	Середня	Висока	Висока
Складність користування	Немає, лише якщо користувач буде використовувати ненадійний пароль, що призведе до витоку інформації.	Складність у розповсюдженні та підтримки [15].	Складність для користування пов’язана з поступовим неузгодженням внутрішніх годинників та апаратного ключа. Тому теоретично можна припустити, що за час розсинхронизації, коли одноразовий пароль може стати многоразовим, хакер може скористатися нагодою [16].	Найлегший спосіб для користувача, тому що дозволяє користувачеві перемикатись між різними додатками без повторної аутентифікації.
Наявність протоколу	HTTP, Forms	SSL/TLS	Forms	SAML, WS-Federation, OAuth, OpenID Connect

Продовження таблиці 1

Метод аут. / Критерія	Аутентифікація за паролем	Аутентифікація за сертифікатом/ключем	Аутентифікація за одноразовим паролем	Аутентифікація за токенами
Сфера використання	WEB-сайти, мережеві служби, авторизація, смартфони, планшети [13]	Може використовуватися в електронній комерції або в будь-який інший сфері, де потрібно машинна аутентифікація або необхідні безпечні з'єднання [17].	Мобільний банк, Web-сайти, OTP-токени [13]	Вхід в додаток через обліковий запис в соціальних мережах, систем накопичувальних знижок до кредитних і дебетових карт, студентських квитків, телефонів стандарту GSM, проїзних квитків[14].
Переваги	Простота і звичність	Надійність	Пароль неможливо використати повторно	Використання один і той самий токен для доступу на різних додатках, простота
Недоліки	Недоліком парольної аутентифікації можна вважати людський фактор, через те що більша частина користувачів використовує ненадійний пароль, також зберігає пароль на листочках.	Недоліком можна вважати тільки складність отримання сертифікату, а також малий термін використання.	Один з недоліків є те, що треба використовувати додаткові засоби для отримання одноразового паролю.	Немає захисту на всі 100%, через те що, можливий сценарій «людини по середині»

3 ТЕХНІЧНА РЕАЛІЗАЦІЯ МЕТОДІВ OAUTH 2.0 І WEBAUTHN

3.1 Реалізація методу OAuth 2.0

Найбільш відомий на даний час протокол аутентифікації за токеном – це OAuth. Він був опублікований у 2012 році, а його ранішня версія 2007 році.

За допомогою OAuth 2.0 користувач дозволяє певним сайту отримати свої закриті дані з соцмереж, але без передачі сайту своїх логінів / паролів. Приведу приклад, коли ви входите на сайті через Facebook, то як раз і надаєте цьому сайту дозвіл отримати з Facebook ваше ім'я, e-mail адресу і інші закриті дані.

Стандарт визначає наступні ролі:

- Resource Owner - користувач, який заходить на MySite і дає йому дозвіл використовувати свої закриті дані з Соцмережі.
- Client (він же MySite) - додаток або інтернет сайт, за допомогою якого користувач і яке взаємодіє з Authorization Server і Resource Server для отримання закритих даних користувача.
- Authorization Server - сервер який перевіряє логін / пароль користувача, він же Соцмережа.
- Resource Server - зберігає закриту інформацію користувача, яку можна отримати за допомогою API. Authorization Server і Resource Server можуть бути суміщені в одну систему [22].

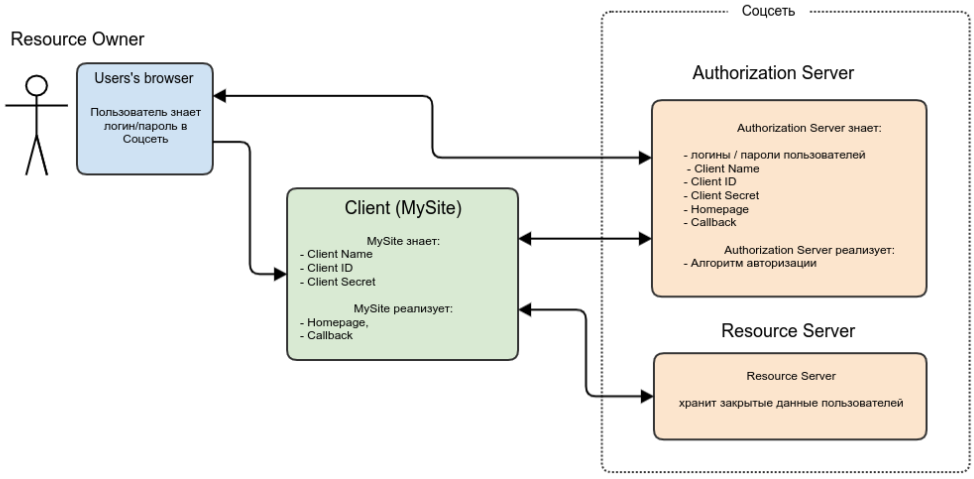


Рисунок 11 – Схематичний вигляд розподілу ролей методу OAuth 2.0

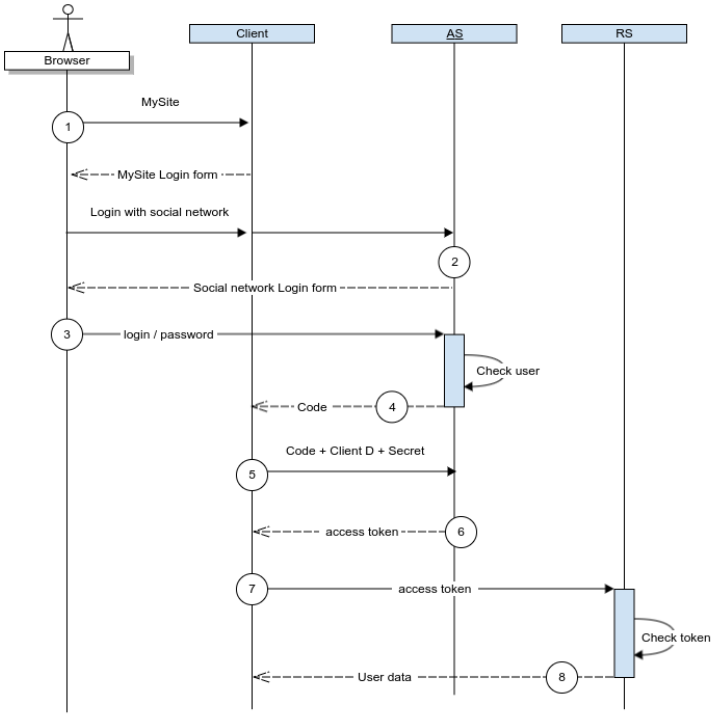


Рисунок 12 – Схематичний вигляд роботи методу OAuth 2.0

3.2 Реалізація методу WebAuthn

Основне завдання стандарту - позбавити користувачів від логінів / паролів і перейти на ідентифікацію за допомогою криптографії з використанням публічних ключів «Public key cryptography». Public key cryptography - це криптографічний концепція, яка використовує пари математично пов'язаних ключів. Private key (закритий ключ) зберігається в безпечному місці у користувача, а public key (відкритий ключ) зберігається і використовується відкрито.

Стандарт дає користувачам можливість ідентифікуватися на сайтах і в додатках за допомогою зовнішніх ключів безпеки (наприклад, USB-ключів) або по відбитку пальця і, згодом, за іншими біометричними даними: особі, сітківці ока [22].

У порівнянні з OAuth 2.0 в WebAuthn додаються ролі:

- Authenticator: зовнішній ключ безпеки (фізичний носій або сканер відбитків пальців), який дозволяє аутентифицировать користувача, використовую різні технології, такі як BlueTooth / NFC / USB. Служить для:
  - Генерації public key credentials (пар відкритих / закритих ключів).
  - Authenticator безпечно зберігає закритий ключ у своїй пам'яті
  - Передає відкритий ключ зовнішнім системам
  - Підписує дані закритим ключем і результат передає зовнішнім системам

Для взаємодії з браузером Authenticator використовує протокол CTAP (Client to Authenticator Protocols).

- Relying Party: виконує ту ж функцію, що і "Authorization Server" в OAuth 2.0, а саме перевіряє ідентифікаційні дані користувача. Тільки в OAuth 2.0 це були логін / пароль, а в WenAuthn - public key credentials.
- User Agent: об'єднує браузер і мережеве додаток, служить для тих же цілей, що і Client в OAuth 2.0, а саме - з одного боку взаємодіє з користувачем і надає йому GUI, а з іншого боку взаємодіє з системами, в яких зберігаються ідентифікаційні дані користувача.

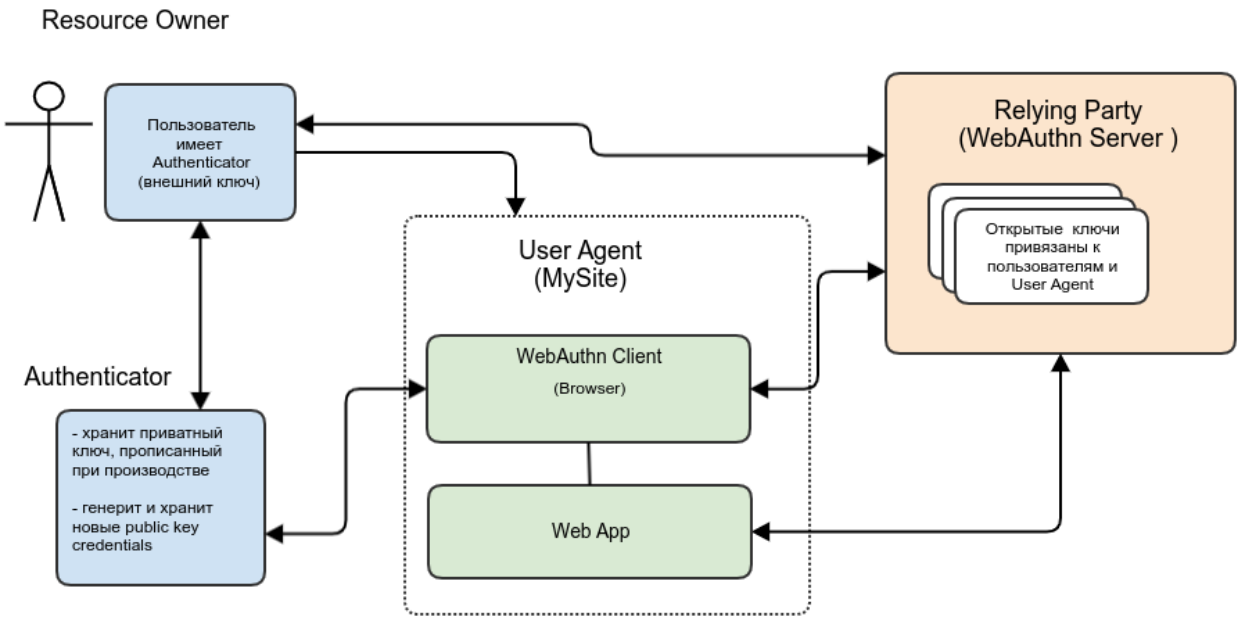


Рисунок 13 – Схематичний вигляд розподілу ролей методу WebAuthn

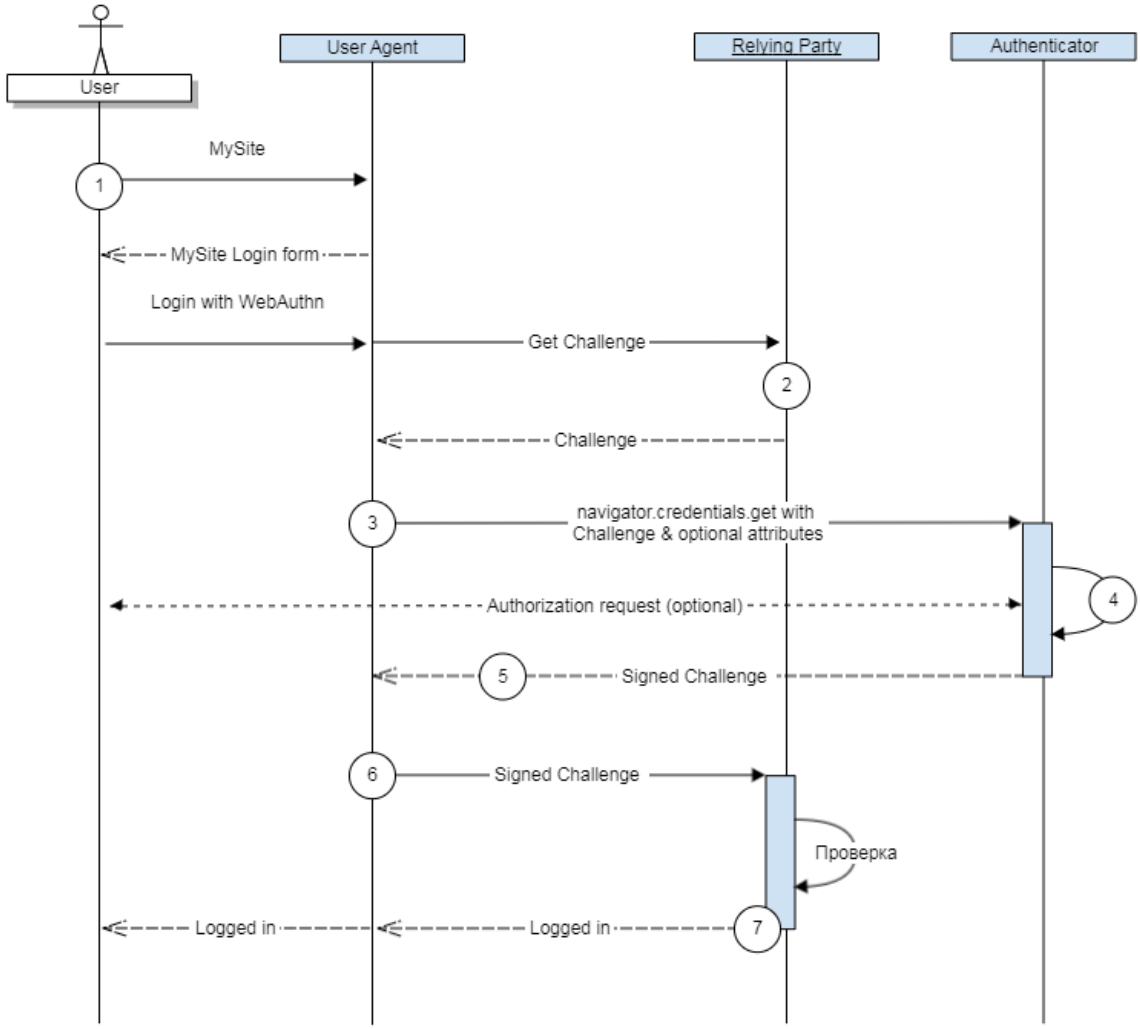


Рисунок 14 – Схематичний вигляд роботи методу WebAuthn

3.3 Висновки

Ми подивились та реалізували два методи аутентифікації – це OAuth 2.0, WebAuthn. І прийшли до такого висновку:

- З'явилася можливість ніколи не зберігати пароль користувача, щоб не компрометувати свою репутацію, а зберігати тільки логіни і зашифровані особисті дані.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Вимоги безпеки при виконанні робіт на робочому місці

Усі давно вже знають, що кожен має право на належні, безпечні і здорові умови праці реалізується через охорону праці як систему правових, соціально – економічних, організаційно – технічних, санітарно – гігієнічних і лікувально – профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я та працездатності людини у процесу трудової діяльності, це нам гарантує Конституція України від 21.11.2002 року №1 [1].

У відповідності до вимог ст. 153 Кодексу законів про працю України та ст. 6 Закону України «Про охорону праці» на всіх підприємствах, в установах, організаціях створюються безпечні і нешкідливі умови праці № 5462-VI від 16.10.2012 [2]. Це свідчить про те, що роботодавець або уповноважений ним орган на всіх підприємствах/установах/організаціях зобов'язаний створити безпечні і нешкідливі умови праці для працівника. Роботодавець, або уповноважений ним орган, зобов'язані впровадити сучасні засоби техніки, для того щоб запобігти виробничого травматизму і забезпечити санітарно – гігієнічні умови, що запобігають виникненню робочих травм та/або захворювань. Власник або уповноважений ним орган не вправі вимагати від працівника виконання роботи в умовах які не відповідають законодавству про охорону праці. Власник або уповноважений ним орган покладається тимчасове проведення інструктажу працівників, або їх навчання з питань охорони праці або протипожежної охорони.

Норма тривалості робочого часу у відповідності до вимог Кодексу законів про працю України [2] нормальна тривалість робочого часу працівника не повинна перевищувати 40 годин на тиждень. Це свідчить про те, що роботодавець не має право вимагати більший робочий день після підписання договорю, який свідчить про час на робочий тиждень.

Робочі місця офісних працівників, обладнані персональними комп'ютерами, повинні відповідати вимогам «Правил охорони праці під час експлуатації електронно – обчислювальних машин», затверджених Наказом Державного комітету України з промислової безпеки, охорони праці та гірничого

нагляду від 26.03.2010 року №65 [3] та «Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно – обчислювальних машин», затверджених постановою Головного державного санітарного лікаря України від 10.12.1998 року №7 (ДСанПіН 3.3.2-007-98) [4]. Ці правила повинні додержуватись на всіх суб'єктах незалежно від форм діяльності, у тому числі на тих, які мають робочі місця, які обладнанні персональними комп'ютерами.

Вимоги щодо розміщення та планування приміщення для роботи з персональним комп'ютером:

- Робочі місця, на яких присутні персональні комп'ютери, заборонено облаштовувати у підвальних або цокольних приміщеннях.
- При обладнанні забороняється використовувати полімерні матеріали, які виділяють шкідливі хімічні речовини.
- Слід приділити увагу до достатнього рівня освітлення робочого місця та звукоізоляції.
- При регулюванні освітленістю слід застосовувати жалюзі.
- В приміщеннях, де використовуються персональні комп'ютери, необхідно щодня виконувати вологе прибирання, з метою недопущення запиленості підлоги та меблів.
- Батареї опалення, водопровідні труби, кабелі із заземленими відкритим екраном, та усі інші заземлені конструкції, які знаходяться в приміщенні, слід надійно завищувати діелектричними щитками/сітками задля недопущення потрапляння людини під напругу.
- В приміщенні необхідно лінії передач мають бути захищені від виникнення короткого замикання, а також від перепадів напруги, що може спричинити збої в роботі, або пожежі.
- Приміщення мають бути оснащені пожежною сигналізацією та вогнегасниками.
- У приміщенні, де одночасно працюють понад 5 комп'ютерів, необхідно обладнати на доступному та помітному місці аварійний резервний вимикач.

Вимоги щодо організації та обладнання робочих місць:

- Площа на одне робоче місце – не менше 5 кв.м.
- Об’єм – не менше 20 куб.м.
- Конструкція робочого місця повинна забезпечувати підтримання оптимальної пози, тобто такої, яка дозволяє працівникові виконувати роботу з мінімальним напруженням тіла.
- За потреби особливої концентрації уваги під час виконання робіт суміжні робочі місця операторів необхідно відділяти одне від одного перегородками висотою 1,5 – 2 м.
- Робочі місця слід розташовувати відносно джерела природного світла (вікон) таким чином, щоб світло падало збоку, переважно зліва.
- Робоче місце повинно відповідати вимогам ергономіки.

Вимоги безпеки під час роботи з комп’ютером:

- Кожного дня перед початком робочого дня необхідно очищати монітор від пилу та інших забруднень
- Після закінчення робочого дня слід виключати усі електронні прибори від електричної мережі.

Після закінчення роботи з використанням персонального комп’ютеру необхідно дотримуватися такої послідовності вимикання обладнання:

- закрити всі активні завдання;
- переконатися у відсутності дискет та дисків у дисководах;
- використавши опцію "Завершення роботи" у меню "Пуск", вимкнути живлення системного блоку;
- вимкнути живлення всіх комп’ютерів;
- вимкнути блок аварійного живлення (за наявності);
- відключити комп’ютер від електромережі, при цьому забороняється тягнути штепсельну вилку за дріт.

Вимоги щодо виникнення аварійних ситуацій необхідно негайно виключити комп’ютер та усі периферійні пристрої від електричної мережі. Що не допускається:

- Виконувати обслуговування/ремонт/налагодження персонального комп’ютеру самому.
- Зберігати на робочому місці, біля комп’ютера, папір або будь-які інші носії інформації, деталі, тощо, якщо вони не використовуються у поточний час.
- Відключати захисні пристрої, самочинно проводити зміни у конструкції комп’ютера.

Працівник під час роботи зобов’язаний:

- виконувати тільки ту роботу, яку йому було доручено;
- підтримувати порядок і чистоту на робочому місці;
- тримати відкритими всі вентиляційні отвори обладнання;
- коректо закрити всі активні завдання у разі припинення роботи з комп’ютером;
- негайно відключити комп’ютером від електричної мережі у разі виникнення аварійної ситуації.

4.2 Шкідливі виробничі фактори на робочому місці

Згідно до Законів України про «Про охорону праці» [2] на всіх підприємствах, в установах, організаціях створюються безпечні і нешкідливі умови праці, нажаль, не усі дотримуються вимог сучасних ергономіки (дивись пункт 1.1), через що виникають професійні захворюваності.

Професійне захворювання – це захворювання, яке було визвано впливом шкідливих умов праці.

Шкідливим називають виробничий фактор, вплив якого, на організм працюючого, може призводити в певних умовах до захворювання або зниження рівня працездатності.

У зв’язку з Кодексом України [1] людина має право на лікувально – профілактичних заходів.

Профілактика професійних захворювань – це система медичних мір і немедичного характеру, направленого на запобігання нещасних випадків на виробництві, зниження ризику відхилень розвитку у стані здоров’я робітника.

Розвиток багатьох професійних захворювань залежить від комплекстної взаємодії факторів і від якості трудового життя.

З 80-х років 20 століття Міністерство Здоров’я України під здоровою людиною розуміє: «Здоров'я - це не тільки відсутність хвороб, а стан фізичного, психічного і соціального благополуччя».

Здоров’я людини формується не тільки під впливом складного комплексу внутрішніх чинників, а ще й зовнішних впливів. На даний час існує формула здоров’я, яку дослідили та визначили вчені устого світу:

Здоров'я 100% = 10% медицина + 20% спадковість + 20% довкілля + 50% образ життя.

Будь-яка професійна діяльність несе в собі певний ступінь ризику професійних захворювань. На думку інших, спеціальність така як програміст та людей інших сидячих професій, нічого складного, не присутня ніяка фізична втома, легка и спокійна праця, але ні. Ця спеціальність дуже малорухома та сидяча професія, яка створює багато передумов для розвитку професійних травм.

Усі знають, що недолік рухових навантажень сучасного програміста знижує активність клітин його організму. Як результат - у них низька фізична витривалість і багато проблем зі здоров'ям.

Найголовніше, що треба дотримуватись працівнику це вимог, щодо створення свого робочого місця:

- Конструкція робочого місця користувача візуально дисплейним терміналом має забезпечити підтримання оптимальної робочої пози.
- Робочі місця з візуально дисплейним терміналом слід так розташовувати відносно світлових прорізів, щоб природне світло падало збоку, переважно зліва.
- При розміщенні робочих столів з візуально дисплейним терміналом слід дотримуватись таких відстаней: між бічними поверхнями ВДТ – 1,2 м; від тильної поверхні одного візуально дисплейним терміналом до екрана іншого – 2,5 м.

- Екран візуально дисплейним терміналом має розташовуватися на оптимальній відстані від очей користувача, що становить 600...700 мм, але не ближче ніж за 600 мм з урахуванням розміру літерно-цифрових знаків і символів.
- Розташування екрана візуально дисплейним терміналом має забезпечувати зручність зорового спостереження у вертикальній площині під кутом +30° до нормальної лінії погляду працюючого.
- Клавіатуру слід розташовувати на поверхні столу на відстані 100...300 мм від краю, звернутого до працюючого. У конструкції клавіатури має передбачатися опорний пристрій (виготовлений із матеріалу з високим коефіцієнтом тертя, що перешкоджає мимовільному її зсуву), який дає змогу змінювати кут нахилу поверхні клавіатури у межах 5...15°.
- При оснащенні робочого місця з візуально дисплейним терміналом лазерним принтером параметри лазерного випромінювання повинні відповідати вимогам ДСанПіН 3.3.2.007- 98 [4].

Основні фактори, які заважають програмісту мати гарне здоров’я:

- пил: він розвиває алергію;
- вплив електромагнітного випромінювання монітора;
- втома очей, навантаження на зір;
- сидяче положення довгий час;
- стрес;

Найпоширеніші захворювання програмістів це:

- остеохондроз, артрит, радикуліт,
- ожиріння, захворювання серцево-судинної системи,
- короткозорість, гіподинамія, синдром зап'ястного каналу, очні захворювання,
- варикоз,
- геморой,
- розсіяний склероз,
- анемія,

- запаморочення, мігрені, захворювання хребта.

Нажаль, існуючі захворювання у працівників які вже в наявності неможна вилікувати, якщо це дуже серйозне захворювання, але для інших працівників можна заради профілактики проводити таку профілактику:

- Кожні 45 хвилин роботи робити відпочинок 15 хвилин;
- За ці 15 хвилин відпочинку, можна зробити гімнастику очей (дивись рисунок 1);
- Встати із-за комп'ютера і просто пройтись, це прибере напруження в м'язах.

Рисунок 14 – гімнастика для очей, заради профілактики захворювань пов'язаних зі зором.

4.3 Дії працівників в аварійний ситуації

Електротравма — це місцеві і загальні пошкодження, що виникають у результаті впливу електричного струму великої сили або розряду атмосферної електрики (блискавки) згідно до порядку надання домедичної допомоги постраждалим при ураженні електричним струмом та блисковкою Наказом Міністерства охорони здоров'я України №398 від 16.06.2014 року [5].

Хворий миттєво втрачає свідомість, відмічається судомне скорочення м'язів, зупинка дихання, різкий розлад серцевої діяльності.

Якщо ураження не призвело до моментальної загибелі та через деякий час свідомість постраждалого відновилася, то в нього визначається головний біль, сонливість, загальна слабкість, млявість, пронос. Місцево відмічаються сліди опіку у вигляді жовтувато-бурих плям та смуг.

Згідно до вищесказаного Наказу Міністерства охорони здоров’я України [4] послідовність дій при наданні домедичної допомоги постраждалим при ураженні електричним струмом та блискавкою не медичними працівниками:

- переконатися у відсутності небезпеки;
- викликати бригаду екстреної (швидкої) медичної допомоги;
- провести огляд постраждалого, визначити наявність свідомості, дихання;
- якщо постраждалий перебуває під дією електричного струму, при можливості припинити його дію: вимкнути джерело струму, відкинути електричний провід за допомогою сухої дерев’яної палиці чи іншого електронепровідного засобу;
- якщо у постраждалого відсутнє дихання, розпочати проведення серцево-легеневої реанімації;
- накласти на місця опіку чисті, стерильні пов’язки;
- при погіршенні стану постраждалого до приїзду бригади екстреної (швидкої) медичної допомоги повторно зателефонувати диспетчеру екстреної медичної допомоги.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Скільки українців користуються Інтернетом і скільки дивляться телебачення: статистика. URL: https://news.24tv.ua/ru/skolko_ukraincev_polzujutsja_internetom_i_skolko_smotrjat_televidenie_statistika_n1285751#:~:text=%D0%93%D0%BE%D1%81%D1%83%D0%B4%D0%B0%D1%80%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%B0%D1%8F%20%D1%81%D0%BB%D1%83%D0%B6%D0%B1%D0%B0%20%D1%81%D1%82%D0%B0%D1%82%D0%B8%D1%81%D1%82%D0%B8%D0%BA%D0%B8%20%D0%BF%D0%BE%D0%B4%D1%81%D1%87%D0%B8%D1%82%D0%B0%D0%BB%D0%B0%2C%20%D1%87%D1%82%D0%BE,%D1%84%D0%B8%D0%B7%D0%BB%D0%B8%D1%86%20%E2%80%93%202%20%D0%BC%D0%B8%D0%BB%D0%BB%D0%B8%D0%BE%D0%BD%D0%B0%20816%20%D1%82%D1%8B%D1%81%D1%8F%D1%87 (Дата звернення: 09.09.2020).
2. Адріанов В.В., Зефіров С.Л., Голованов В.Б., Голдуев Н.А. Обеспечение информационной безопасности бизнеса.Москва: Альпина Паблишер, 2011. 373 с.
3. Огляд способів і протоколів аутентифікації в веб-додатках. URL: https://habr.com/ru/company/dataart/blog/262817/ (Дата звернення: 09.09.2020).
4. http basic authentication або http авторизація. URL: https://web-programming.com.ua/http-basic-authentication-ili-http-avtorizaciya/ (Дата звернення: 15.09.2020).
5. Аутентифікація за допомогою форм. URL: https://professorweb.ru/my/ASP_NET/security/level2/2_1.php (Дата звернення: 15.09.2020).
6. Аутентифікація на основі сертифікатів. URL: http://iptcp.net/autentifikatsiya-na-osnove-sertifikatov.html (Дата звернення: 15.09.2020).
7. Евтеев Д., Гордейчик С., Как я перестал бояться токенов и полюбил одноразовые пароли. Современные технологии безопасности. 2008. №5. С. 2-5.
8. WebAuthn в реальной жизни. URL: https://habr.com/ru/company/mailru/blog/489270/ (Дата звернення: 15.09.2020).
9. Аутентифікація і авторизація в мікросервісних додатках. URL: https://dataart.ua/news/autentifikatsiya-i-avtorizatsiya-v-mikroservisny-h-prilozheniyah/ (Дата звернення: 20.09.2020).
10. OAuth 2.0 простою і зрозумілою мовою. URL: https://habr.com/ru/company/mailru/blog/115163/ (Дата звернення: 20.09.2020).
11. Ілюстроване керівництво по OAuth і OpenID Connect. URL: https://temofeev.ru/info/articles/illyustrirovannoe-rukovodstvo-po-oauth-i-openid-connect/ (Дата звернення: 20.09.2020).

12. Що таке SAML аутентифікація і кому вона потрібна?. URL: https://habr.com/ru/company/gemaltorussia/blog/322316/ (Дата звернення: 20.09.2020).
13. Мартынова, Л. Е. Исследование и сравнительный анализ методов аутентификации / Л. Е. Мартынова, М. Ю. Умницын, К. Е. Назарова, И. П. Пересыпкин. — Текст : непосредственный // Молодой ученый. — 2016. — № 19 (123). — С. 90-93.
14. Сучасні методи аутентифікації: токен і це все про нього!. URL: https://www.aladdin-rd.ru/company/pressroom/articles/sovremennye_metody_autentifikacii_token_i_eto_vse_o_nem (Дата звернення: 05.10.2020).
15. Аутентифікація на основі сертифікатів. URL: https://sites.google.com/site/galihar23/bezopasnost-os/bezopasnost/autentifikacianaosnovesertifikatov (Дата звернення: 08.10.2020).
16. Аутентифікація на основі одноразового пароля. URL: http://iptcp.net/autentifikatsiya-na-osnove-odnorazovogo-parolya.html (Дата звернення: 08.10.2020).
17. Аутентифікація за сертифікатом. URL: http://csaa.ru/autentifikacija-po-sertifikatam/ (Дата звернення: 10.10.2020).
18. Клієнтський SSL сертифікат: для чого використовується. URL: https://www.leaderssl.ru/articles/211-klientskiy-ssl-sertifikat-dlya-chego-ispolzuetsya (Дата звернення: 19.10.2020).
19. Введення в OAuth 2. URL: https://www.digitalocean.com/community/tutorials/oauth-2-ru (Дата звернення: 19.10.2020).
20. Як користувачі сприймають різні методи аутентифікації. URL: https://habr.com/ru/post/344406/ (Дата звернення: 24.10.2020).
21. Найпопулярніші і погані паролі 2020 року. URL: https://www.ixbt.com/news/2020/11/19/samye-populjarnye-i-plohie-paroli-2020-goda.html (Дата звернення: 24.10.2020).
22. Сучасні стандарти ідентифікації: OAuth 2.0, OpenID Connect, WebAuthn. URL: https://m.habr.com/ru/post/491116/ (Дата звернення: 24.11.2020).
23. Право на належні, безпечні і здорові умови праці реалізується через охорону праці як систему правових, соціально – економічних, організаційно – технічних, санітарно – гігієнічних і лікувально – профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я та працездатності людини у процесу трудової діяльності, це нам гарантує Конституція України від 21.11.2002 року №1
24. Вимог ст. 153 Кодексу законів про працю України та ст. 6 Закону

України «Про охорону праці» на всіх підприємствах, в установах, організаціях створюються безпечні і нешкідливі умови праці № 5462-VI від 16.10.2012

25. Наказом Державного комітету України з промислової безпеки, охорони праці та гірничого нагляду від 26.03.2010 року №65

26. «Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно – обчислювальних машин», затверджених постановою Головного державного санітарного лікаря України від 10.12.1998 року №7 (ДСанПіН 3.3.2-007-98)

27. Наказом Міністерства охорони здоров'я України №398 від 16.06.2014 року